How to Make Business Intelligence Agile: The Agile BI Actions Catalog

Robert Krawatzeck
Chemnitz University of Technology
robert.krawatzeck@wirtschaft.tu-chemnitz.de

Barbara Dinter
Chemnitz University of Technology
barbara.dinter@wirtschaft.tu-chemnitz.de

Duc Anh Pham Thi
Chemnitz University of Technology
duc-anh.pham-thi@s2011.tu-chemnitz.de

Abstract

The need to develop Business Intelligence (BI) systems which are able to react to unforeseen or volatile requirements in a given time frame results from increasingly complex and dynamic organizational environments. This adaptation capability of BI systems is usually referred to as ‘BI agility’. Although a consensus within literature exists about the demand for agile BI systems, a structured overview on how such systems can be developed is missing. Closing this research gap, the paper at hand presents a catalog of Agile BI actions which is based on a comprehensive literature review. It includes 21 assessed actions suitable to increase the agility of BI systems. Therefore the catalog provides benefits to scientists (identification of research areas) as well as practitioners (identification of sophisticated actions for implementation). It consists of the four action categories ‘principles’, ‘process models’, ‘techniques’, and ‘technologies’.

1. Introduction

Due to increasingly complex and dynamic markets, most enterprises face the challenge of rapid and frequent adaptations of their information systems (IS) landscape in order to remain competitive [36]. Same is in particular true for Business Intelligence (BI) systems which have also to be adapted to the dynamic environment in a rapid manner [9, 23, 44]. BI encompasses all processes and systems that are dedicated to the systematic and purposeful analysis of an organization and its competitive environment [29, 40].

The capability of BI systems to adapt to environmental changes adequately is called ‘BI agility’ [45]. However, a broad range of actions is required to fulfill the demand for BI agility. To the best of our knowledge no comprehensive and systematic overview of these ‘Agile BI actions’ exists so far. It could guide organizations in the systematic and stepwise realization of agile BI systems.

Therefore, the paper at hand aims at filling this research gap by providing the respective overview of actions which improve the agility of BI systems. A so-called Agile BI actions catalog is developed by means of a comprehensive review and consolidation of previous Agile BI literature. In addition to the action list, the actions are examined and assessed according to their level of implementation. In doing so, the Agile BI action catalog provides additional benefits from a scientific as well as a practitioners’ point of view: Scientists can easily identify actions which are missing or should be improved (low implementation level) and practitioners can easily identify actions which have been tested and proven to work (high level of implementation).

Therefore, the remainder of the paper is structured as follows. Section 2 investigates the status quo of Agile BI. The subsequent Section 3 introduces the research methodology and a categorization framework used for the catalog. Section 4 outlines in detail the results of the literature review and the categorization of identified Agile BI actions. The results are analyzed and discussed in Section 5. Finally, we present a conclusion and provide an outlook to future research (Section 6).

2. Status quo of Agile BI

As an extensive literature review, classification, and discussion of Agile BI related papers is in the focus of the remainder of this article, we keep the state of the art overview rather short and focus on the Agile BI understanding and the need for an Agile BI actions catalog.

We follow a common understanding of BI which describes a broad category of applications, technologies, architectures, and processes for gathering, storing, accessing, and analyzing operational data to provide business users with timely competitive
information to enable better insights for operational and strategic decision-making [29, 40].

Although various definitions for agility exist [e.g. 32, 35, 42], they commonly are based on the well-known ‘Manifesto for Agile Software Development’ [7] and emphasize the ability to respond to unforeseen changes quickly [45].

The application of agile principles to BI systems gains increasing relevance. Consequently, the question how current BI systems can be flexibly adapted to frequently changing requirements caused by the dynamic business environment is subject to many articles [23]. As a matter of fact, recent research contributions have focused on the usage of agile process models such as Scrum or eXtreme Programming [12, 20, 21, 44]. These agile process models are without a doubt an important means for developing BI systems in an agile manner and hence to improve the development process in terms of time and flexibility. Nonetheless, in the context of BI understandings of agility exist that are different and broader than the exclusive adaptation of agile process models as known from software engineering (SE) [27].

The variety of definitions is illustrated by [23] who provide an overview of eleven agility definitions in an IS or BI context. To overcome the curse of different understandings of agility and BI and to establish an understanding shared by academics and practitioners, the German Chapter of The Data Warehouse Institute (TDWI) developed and proposed a definition for ‘Agile BI’ [24, 45]. It was a joint effort of 30 BI consultants, vendors, users, and researchers. Based on the definition of ‘BI agility’ as a characteristic of a BI system describing the ability “to react to unforeseen or volatile requirements regarding the functionality or the content of a BI solution in a given time frame” [45], ‘Agile BI’ has been defined as all actions undertaken by an organization to achieve or improve this characteristic [24].

Due to the practitioner-oriented origin this definition has been complemented by a framework which distinguishes four action categories for Agile BI [24]:

**Principles:** A principle can be defined as a rule and assumption derived from extensive observation and evolved through years of experience [39]. Within the SE domain the KISS principle – Keep it simple, stupid – is a well-known design principle.

**Process models:** A process model provides guidance to coordinate and control different tasks systematically which must be performed in order to achieve a specific (project) goal [39]. Within the SE domain the waterfall model is a well-known example for a process model.

**Techniques:** A technique is a way or style of carrying out a particular task. A specific data modeling technique might serve as an example.

**Technologies:** A technology is the application of knowledge to describe technical procedures for problem solving, e.g. in the form of paradigms, specific tools or systems, such as the paradigm of service-oriented architectures (SOA).

As already mentioned, many papers stress the importance and impact of Agile BI. Besides the argumentation for the need of means to increase BI agility [45], first attempts for indicators measuring the success of Agile BI actions have already been suggested [6]. However, a comprehensive and systematic overview of such potential actions is still missing.

The obvious approach to transfer and adapt well-known agile actions from the SE context to BI exhibits major challenges: BI systems are different – that’s why the concept of dedicated Agile BI has been developed and why generic agile means not necessarily can be used one to one in the BI context [24]. Depending on the actions, a corresponding adaptation might become costly and complex, if not even impossible. Due to broad range of potential “classic” agile actions we consider a systematic overview of already known and elaborated Agile BI actions in previous literature (and the subsequent identification of the missing ones) as a good starting point to support organizations on their way to Agile BI. The value of such an overview increases if it based on a framework and on a consolidated and approved terminology. For the remainder of the paper we derive such a framework and classify Agile BI papers accordingly.

### 3. Research methodology

As motivated in the introduction, our research goal comprises the identification, structuring, and assessment of publications which present and discuss actions to increase the BI agility. Since this goal corresponds to the first and second purpose of a literature review as identified by Schwarz et al. [38] (cf. Figure 1), we regarded the research methodology of a literature review in IS as appropriate and conducted it according to [17].
After summarizing prior research about Agile BI actions we critically examined the publications by evaluating the identified BI actions on the basis of their implementation levels. Thereby, shortcomings and knowledge gaps with regard to the evaluated actions were identified. The respective implementation levels and their description are listed in Table 1.

Table 1. Implementation levels of Agile BI actions in literature

<table>
<thead>
<tr>
<th>Implementation level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level 1</td>
<td>The Agile BI action is presented as a potential solution to increase BI agility.</td>
</tr>
<tr>
<td>Level 2</td>
<td>The Agile BI action is presented and its potential to increase BI agility is validated by argumentation and/or theory.</td>
</tr>
<tr>
<td>Level 3</td>
<td>The Agile BI action and its ability to increase BI agility has been tested or proven in a real project.</td>
</tr>
</tbody>
</table>

Webster and Watson [42] propose a concept-centric approach to structure a literature review since an author-centric approach fails to synthesize the literature. We followed this suggestion and used a concept matrix in this literature review which consists of the aforementioned action categories ‘principles’, ‘process models’, ‘techniques’, and ‘technologies’ in combination with the implementation levels (cf. Table 1). This concept matrix establishes the Agile BI actions catalog classification framework as shown in Figure 2. It consists of twelve potential fields from ‘P1’ to ‘TL3’.

Figure 1. Purpose of review articles [38]

Figure 2. Agile BI actions catalog categorization framework

The process we used to conduct the literature review follows the method according to Fettke [17] which consists of the following steps: (1) problem formulation, (2) search for literature, (3) evaluation of literature, (4) analysis and interpretation, and (5) presentation.

4. Development of an Agile BI actions catalog

4.1. Step 1: Problem formulation

For a detailed description of the research problem we refer to Sections 1 and 2.

4.2. Step 2: Search for literature

We conducted an advanced search for publications that explicitly address agility or agile development issues in the context of data warehousing (DWH) or BI within title, abstract, or keywords. Therefore we operationalized and used the following search query:

(“agile” OR “agility” ) AND 
(“data warehouse” OR “data warehousing” OR “DWH” OR “business intelligence” OR “BI” ).

In order to identify relevant papers we restricted the search to the Top 20 journals ranked by the Association for Information Systems (AIS). Scientific databases used to gain access to the journals’ full-text articles include Scopus, ScienceDirect, and Web of Science.

1 http://aisnet.org/general/custom.asp?page=JournalRankings
As a matter of fact, by May 2014 no matches were found in these 20 AIS journals. Therefore, the research was extended to all journals and conference proceedings included in the three aforementioned scientific databases. The extended search resulted in 229 articles.

4.3. Step 3: Evaluation of literature

In this step, duplicates were eliminated. Also, we checked title and abstract of all remaining articles and removed papers with inappropriate topics. Only those articles were selected which address agility in the context of DWH/BI. Consequently, articles focusing on agility in general or on other topics were excluded. This step resulted in 78 articles.

4.4. Step 4 & 5: Analysis, interpretation and presentation

Each article was carefully reviewed. Due to the search terms used, we received two kinds of articles addressing both, agility and BI: (1) Articles that present and explicitly discuss actions to support or increase BI agility and (2) articles that point out the contribution of BI to increase the overall organizational agility (e.g. [9, 26, 31]).

With respect to our research goal, we focused our analysis on the former articles (1). In total, 19 articles could be identified suggesting actions which increase the agility of BI systems. Those articles were analyzed and classified according to our framework. The results of this step are presented in the paper at hand (conforms to Step 5 of the literature review method by Fettke [17]).

Remarkably, no significant literature about Agile BI actions has been published before 2007 (cf. Figure 2). In other words, six years have passed from the publication of the Manifesto for Agile Software Development in 2001 [7] to the publication of the first Agile BI action.

In the following, we present the allocation of the articles according to the action categories ‘principles’, ‘process models’, ‘techniques’ and ‘technologies’. In addition, the content of each article is briefly sketched for a better overview.

4.4.1. Principles. Three articles could be identified that mention principles in the context of BI agility. These articles are listed in Table 2, sorted by publication year and implementation level.

<table>
<thead>
<tr>
<th>Ref.</th>
<th>Year</th>
<th>Action</th>
<th>Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>[18]</td>
<td>2011</td>
<td>Various principles of agile software development methodology, including: • Twelve principles of agile methodologies applied to data warehousing • Six architecture principles for an agile data warehouse • Six best practices for agile DWH/BI projects</td>
<td>2</td>
</tr>
<tr>
<td>[19]</td>
<td>2011</td>
<td>Six principles, condensed from principles of seven different software development methodologies</td>
<td>3</td>
</tr>
<tr>
<td>[10]</td>
<td>2012</td>
<td>Three agile data warehouse key principles and three technical practices</td>
<td>3</td>
</tr>
</tbody>
</table>

Notably, Agile BI principles were always mentioned in combination with agile process models. This complies with the understanding of ‘process models’ by Goede [18]. According to Goede, process models “give practical guidelines for applying principles”.

Goede [18] compares the agile software development methodology (ASDM) principles and Scrum by conducting a study with a group of students. The students applied the ASDM principles and Scrum by developing a data warehouse. As a consequence, ASDM turned out to be fairly suitable for data warehouse projects from the students’ point of view [18]. Since there is a lack of practical evidence, we assign Goede’s findings to level 2 on the basis of our prior definition.

By contrast, Golfarelli et al. [19] suggest six principles which the authors deduced from seven different software development methodologies. The
consolidated principles served as a starting point for
the development of a process model (Four-Wheel-
Drive; 4WD) suitable for data warehouse development
in an agile manner. Moreover, the principles as well as
4WD were applied and elaborated during two practical
projects. Due to this fact, we rate the actions suggested
by Golfarelli et al. as level 3.

Bunio [10] follows another approach and describes
the lessons learned during an agile data warehouse
redevelopment project. He shared three – successfully
and unsuccessfully used – agile data warehouse key
principles and technical practices so they can be
applied and evaluated in other projects. Since the
principles were developed and applied in practice we
assign these actions to level 3.

4.4.2. Process models. The classification reveals six
articles dealing with process models. Two of them –
[18] with ‘Scrum’ and [19] with ‘4WD’ – were already
introduced in Section 4.4.1 (Agile BI principles). The
following table includes all identified Agile BI process
models.

<table>
<thead>
<tr>
<th>Ref.</th>
<th>Year</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>[34]</td>
<td>2011</td>
<td>Scrum &amp; XP</td>
</tr>
<tr>
<td>[18]</td>
<td>2011</td>
<td>Scrum</td>
</tr>
<tr>
<td>[19]</td>
<td>2011</td>
<td>Four-Wheel-Drive (4WD)</td>
</tr>
<tr>
<td>[24]</td>
<td>2013</td>
<td>Scrum</td>
</tr>
</tbody>
</table>

Scrum [37] is one of the best known representatives
of the agile methodology. Dasgupta and Vankayala
[15] use Scrum as a process model to develop and
maintain an agile BI system. The effect of Scrum on BI
agility was proven by an implementation in a project.
Therefore, this paper was assigned to level 3.

As mentioned in Section 4.4.1, also Goede [18]
tested the suitability of Scrum in a case study among
students. As a result, Goede’s contribution relates to
level 2.

Hughes [24] examines Scrum and the underlying
methods in the context of data warehouse
development. According to Hughes Scrum can be used
as a process model for data warehouse development
and for the optimization of teams by specifying team
roles which allow DW/BI project teams to better meet
the challenges of data integration projects. It remains
unclear, to which extend Hughes’ adaptation of Scrum
for Agile BI has been tested or applied in real
enterprises. Therefore this Agile BI action is rated as
level 2.

Rehani [34] compares the agile and the waterfall
methodologies for Agile BI. As examples for agile
methodologies, Rehani mentions Scrum and XP
eXtreme Programming) and their potential impact on
increasing agility in BI [34]. The article provides a
generic overview of agile methodologies and their
representatives. Consequently Rehani’s article is
assigned to level 1.

Apart from Scrum, there are other examples of
agile methodologies such as the Adaptive Software
Development approach (ASD). [3] developed the
ASD-DM methodology by applying ASD in the BI
context. According to Alnoukari et al. [3], this process
model fits best in a very dynamic business
environment in which requirements are constantly
changing and volatile. The authors’ theory was proven
during a practical project – which leads to a rating as a
level 3 action.

Another process model that seems to be suitable in
case of uncertain requirements is the Four-Wheel-
Drive (4WD) [19]. Golfarelli et al. regard the 4WD as
an action to increase BI agility by following six
principles condensed from seven software development
methodologies (cf. Section 4.4.1). As mentioned in
Section 4.4.1, it was applied in project and corresponds
to level 3.

4.4.3. Techniques. Table 4 gives an overview of six
Agile BI techniques which we could identify in our
review.

<table>
<thead>
<tr>
<th>Ref.</th>
<th>Year</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>[35]</td>
<td>2010</td>
<td>Anchor Modeling</td>
</tr>
<tr>
<td>[12]</td>
<td>2010</td>
<td>Class Object Value Element Relationship (COVER) data model</td>
</tr>
<tr>
<td>[1]</td>
<td>2011</td>
<td>MapReduce to build cubes</td>
</tr>
<tr>
<td>[20]</td>
<td>2012</td>
<td>Optimal sprint planning through optimization</td>
</tr>
</tbody>
</table>

The technique of MAD analysis skills (Magnetic,
Agile, Deep) differs from traditional data warehousing
and BI which requires the cleansing of data before it is
loaded into the data warehouse. Cohen et al. [13]
propose MAD skills as a new different approach to build a data warehouse. First, data warehouses can only meet the challenges of the dynamic business environment by being “magnetic” which means attracting all data sources regardless of their data quality. Second, traditional data warehouses exhibit a fairly long design and planning phase. Given the complexity and the amount of data sources, data warehouses have to be more agile by allowing analysts to “ingest, digest, produce and adapt data quickly” [13]. In order to do so, analysts have to study enormous datasets and they have to run highly sophisticated statistical methods. This requires a modern data warehouse that serves as a comprehensive data repository as well as a complex algorithmic runtime engine. The design and implementation of a MAD data warehouse is formulated as a result from extensive discussion, but has not been realized and tested in a project and/or evaluated otherwise. Therefore this technique is assigned to level 1.

Rönnbäck et al. [35] present Anchor Modeling as a technique to model data warehouses which are easier to evolve. Anchor Modeling allows reliable and flexible data model changes since changes are only allowed in the form of extensions. Therefore backward compatibility is guaranteed at any time. Rönnbäck et al. proof the outperformance of Anchor Modeling in comparison to traditional modeling techniques by conducting several lab experiments but not yet in a real-world project. As a result, this contribution corresponds to level 2.

Besides database modeling the issue of storing and accessing necessary data is very important as well. Thus, Cheng and Yesha [12] propose the COVER model (Class Object Value Element Relationship) as a solution for a better XML data management. On the one hand, it provides simple and integrated data structures for storing data. On the other hand, it preserves the flexibility and agility of XML schemas by utilizing a generic relational data model which allows fast data retrieval. The COVER model performance was tested in an experiment. Accordingly the paper is rated as level 2.

Another technique to improve the agility of data warehouses is introduced by Abelló et al. [1]. The authors propose MapReduce as a technique to optimize the data storing and extraction issue. MapReduce is a programming model composed of two separate procedures which is utilized as an agile mechanism to deploy cubes in ad-hoc data marts [1]. The performance of the suggested deployment technique was tested within an experiment (level 2).

According to Akkaoui et al. [2] ETL (extract, transform, and load) process designers have to be aware of business processes in order to identify which data is required and how to include it in the data warehouse. For this purpose, the authors developed a BPMN (Business Process Modeling and Notation)-based metamodel for the design of conceptual models of ETL processes. In detail, the authors model by means of BPMN ETL processes that can be mapped to local models and implemented vendor-independently. Furthermore, the BPMN-based metamodel provides the basis for ETL processes to interact with other applications and systems to get real-time data. The metamodel has been tested by experiment only. Therefore, it is assigned to level 2.

In contrast to the aforementioned articles, Golfarelli et al. [20] investigate a Scrum-specific issue - the sprint planning problem. Although there are tools supporting agile project management, no support for an optimal sprint planning is provided. The effectiveness of the sprint planning phase depends on the team experience to estimate the requirements as accurate as possible and on the capability of taking several variables and constraints into account. Golfarelli et al. formulated this issue as an optimization problem whose complexity increases with the project size. Using the IBM ILOG CPLEX Optimizer Golfarelli et al. show that the so called multi-knapsack problem can be solved and can support an optimal sprint planning (implementation level 3).

4.4.4. Technologies. The Agile BI actions assigned to the action category ‘technologies’ are depicted in the following Table 5.

<table>
<thead>
<tr>
<th>Ref.</th>
<th>Year</th>
<th>Action</th>
<th>Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>[4]</td>
<td>2008</td>
<td>Creating an agile infrastructure through MDM, SOA, CEP and BI</td>
<td>1</td>
</tr>
<tr>
<td>[26]</td>
<td>2010</td>
<td>Modeling agile business processes through SOA, BPM, MDM and CEP</td>
<td>1</td>
</tr>
<tr>
<td>[40]</td>
<td>2011</td>
<td>BPM-SOA framework</td>
<td>2</td>
</tr>
<tr>
<td>[25]</td>
<td>2012</td>
<td>Data Virtualization</td>
<td>2</td>
</tr>
<tr>
<td>[8]</td>
<td>2013</td>
<td>Spatial data warehouse (SWD) and spatial on-line analytical processing (SOLAP)</td>
<td>2</td>
</tr>
<tr>
<td>[44]</td>
<td>2013</td>
<td>Agile ETL tool (Microsoft specific)</td>
<td>2</td>
</tr>
</tbody>
</table>

Andreescu and Mircea published two articles that deal with the potential of combining different software systems and architectural paradigms to improve agility in BI. In the 2008 paper [4] the authors suggest to combine service-oriented architectures (SOA), master data management (MDM), and complex event processing (CEP) to create an agile infrastructure of
BI. Later, Andreescu and Mircea extended their approach with the aspect of business process management (BPM) which provides necessary information to improve workflow and streamline business processes [26]. In both articles the combining approach was merely formulated theoretically and did not undergo a theoretical or practical elaboration. As a consequence, the articles exhibit the implementation level 1.

The BPM-SOA framework, as mentioned by Wang and Lee [40], combines the business-driven factors (BPM) as well as the IT-driven aspects (SOA). The framework is conceptually formulated and merely experimentally tested. Therefore, it is rated with level 2.

According to van der Lans Data Virtualization is a technology that “offers data consumers an unified, abstracted and encapsulated view for querying and manipulating a heterogeneous set of data stores” [25]. In other words, Data Virtualization provides an individualized view for data consumers that hides irrelevant technical details and particularly concentrates on relevant data from multiple data stores. Data Virtualization and its impact on BI agility have not been evaluated practically so far (level 2).

Bianchi et al. [8] refer to spatial data warehouses (SDW) and spatial on-line analytical processing (SOLAP) as technologies for agile analytical query processing. A SDW is a dimensional database that stores conventional as well as spatial data. Since Bianchi et al. [8] only refer to the technology of SDW and have tested the performance of different SOLAP tools in an experiment, this action is also assigned to level 2.

Xavier and Moreira [44] introduce agile ETL which simplifies the consolidation of information in data warehouses. This mechanism of agile ETL creates, controls, and monitors processes of extraction, transformation, and loading of data to provide a faster response either in creating or monitoring processes. Agile ETL should be achieved by using different tools such as Microsoft Integration Services, Microsoft Windows Service Applications, Microsoft SQL Server, Windows PowerShell, and Microsoft SQL Server Analysis Services. The authors conducted a performance test in an experiment (implementation level 2).

All in all, it is salient that Agile BI technologies focus on the aspect of providing an agile architecture within BI solutions and on the feasibility of performing dynamic queries to increase BI agility.

Our results will be discussed in detail in the following section.

5. Discussion

The comprehensive literature review revealed 21 Agile BI actions with different levels of implementation. Figure 3 shows the distribution of those actions according to our classification framework. Two insights became obvious: First, we were unable to find Agile BI principles which were developed independently from practical projects (Field ‘P1’ contains no actions) which approves the definition of principles as assumptions derived from extensive observation and evolved through years of experience [39]. Second, the field ‘TL3’ (Agile BI technologies that have been approved within real world settings) remains a research opportunity, since the potential to increase BI agility by means of current Agile BI technologies has been shown through argumentation and/or theory at best (cf. Figure 3).

Figure 3. Number of identified Agile BI actions within the categorization framework

However, the majority of the identified Agile BI actions can be assigned to implementation level 2 (eleven out of 21). Hence, within literature more conceptually formulated or experimentally tested actions exist currently than actions that actually have undergone a practical implementation or evaluation. Consequently, the effectiveness of those actions should be applied in a practical project to either proof or falsify their impact on increasing BI agility as claimed. By contrast, only six of the Agile BI actions identified have been applied within a real world setting.

Analyzing the action category ‘technologies’ in more detail reveals another observation: All technology actions identified are so called back-end technologies, i.e. programs, not directly accessed by the user, which perform a specialized function on behalf of a main software system. Front-end technologies which are directly accessed by the user
are not subject to Agile BI actions so far. This raises the question if front-end technologies suitable to increase BI agility exist. Within other domains such technologies have already been proposed, e.g. Pankaj et al. [31] who suggest Business Dashboards as a technology to archive enterprise agility.

The action category ‘techniques’ currently includes two types of techniques: (1) techniques that describe how specific BI tasks, such as ETL process modeling, could be performed and (2) techniques which describe how specific agile methodology tasks, like the planning of an iteration, can be optimized. With only one representative of the latter and many BI-specific tasks currently remaining uncovered, the largest potential for future research might be in this category. This research could focus on an agile “spike” solution covering all layers of a BI architecture.

The action category ‘process models’ is the only one including an action that is suggested by several authors. The enormous popularity of ‘Scrum’ within the SE domain becomes also evident within Agile BI.

The actions within the category ‘principles’ are in most cases strongly related to ‘process models’. All actions have their origin in the SE domain and have been transferred to the BI context.

Besides the identification of missing actions (e.g. for Agile BI front-end technologies), we have found actions which could not directly be classified within our concept matrix. For example, we identified suitable actions which suggest different cross-sectional management tasks or organizational units and therefore could not be classified using the four aforementioned categories, e.g. the action ‘Data Quality through Agile Data Governance’ [29]. As an additional example Zimmer et al. [5] propose a Business Intelligence Competence Center (BICC) that accompanies the whole BI development process. Hence, an established BICC has a supporting role within a process model and can be labeled as an organizational control mechanism, but is not a ‘principle’, ‘process model’, ‘technique’ or ‘technology’ itself. To overcome this issue, we suggest that the definition of potential Agile BI categories as presented by [24] should be extended by a fifth category ‘organization’ (or a similar naming).

Another type of articles which could not be classified is presented by Mircea et al. [28]. Mircea et al. present Cloud BI as an effective solution to improve an organization’s agility. Although the authors claim that cloud computing increases the organizational agility, an explicit statement about the suitability of cloud computing for Agile BI is missing. Therefore, no classification was possible and future research should consider cloud-based solutions as an effective way to implement Agile BI.

Nevertheless, our catalog gives an overview of Agile BI actions as suggested in the literature so far and allows a comparison of those actions among their implementation level. To gain additional benefit from this overview from a practitioner’s point of view, further research should identify in particular efficient combinations of the Agile BI actions, e.g. which technologies support which techniques or process models best. In general, an assessment of the actions according to their effectiveness and efficiency (e.g. by means of agility indicators as introduced by [6]) might provide further guidance for Agile BI.

To keep our catalog up to date or extend it over time, we suggest (1) to repeat our chosen approach of review from time to time and (2) to transfer additional agile actions known from SE to BI systematically to fill obvious gaps in the catalog.

6. Conclusion and further work

Based on a comprehensive literature review, the paper at hand presents an Agile BI actions catalog. The catalog includes 21 Agile BI actions, structured in the four action categories ‘principles’ (three actions), ‘process models’, ‘techniques’ and ‘technologies’ (six actions each).

In addition to the structured collection of actions – which enables a quick overview about the current Agile BI body of knowledge – each catalog item is described briefly and has been assessed regarding its level of implementation (in a range from ‘an idea only’ to ‘applied within a real world setting’). The catalog can be used by scientific community as well as practitioners to identify promising research areas or sophisticated actions for implementation, respectively.

As outlined in the discussion section, the Agile BI actions catalog constitutes a promising starting point for further Agile BI research. For instance, many identified Agile BI actions are not developed to their full extent and/or have not been evaluated yet.

Besides the assessment of the implementation level within the categorization framework, an additional dimension allowing the comparison of the actual degree of impact of an Agile BI action on BI agility is missing and remains subject to further work. This missing dimension could possible introduced by an evaluation of all actions identified against one of the agility indexes known from SE, e.g. the Agility Index Measurements (AIM) [9], the Agility Measurement Index (AMI) [16], the Agile Maturity Model Integration (AMMI) [22], or the BI agility indicators suggested by [6].
In the long run, these steps – starting with the Agile BI actions catalog – may prepare the way for a dedicated Agile BI maturity model.

7. References


